Command: **elle – ELLE Looks Like Emacs**
Syntax: **elle** *file* [*file2*]
Flags: (none)
Example: elle file.c # Start the editor

ELLE (ELLE Looks Like Emacs) is an Emacs clone for MINIX. ELLE is not full Emacs but it has about 80 commands and is quite fast.

Key bindings

Mined only has a small number of commands. All of them are either of the form CTRL-x or are on the numeric keypad. Emacs, in contrast, has so many commands, that not only are all the CTRL-x commands used up, but so are all the ESC x (escape followed by x; escape is not a shift character, like CTRL). Even this is not enough, so CTRL-X is used as a prefix for additional commands. Thus CTRL-X CTRL-L is a command, and so is CTRL-X K. Note that what is conventionally written as CTRL-X K really means CTRL-X k. In some contexts it is traditional to write CTRL-X as ˆX. Please note that they mean the same thing.

As a result, many Emacs commands need three or four key strokes to execute. Some people think 3-4 key strokes is too many. For this reason, Emacs and ELLE allow users to assign their own key bindings. In ELLE this is done with ‘‘user profiles.’’ A user profile is a file listing which function is invoked by which key stroke. The user profile is then compiled by a program called ellec into binary form. When ELLE starts up it checks to see if a file .ellepro.b1 exists in $HOME. If it does, this file is read in and overrides the default bindings.

A user profile that simulates the *mined* commands fairly well is provided. Its installation is described later. If you have never used Emacs, it is suggested that you use the *mined* profile. If you normally use Emacs, then do not install the *mined* profile. You can also make your own using *ellec*. There is no Mock Lisp.

ELLE has a character-oriented view of the world, not a line oriented view, like *ed*. It does not have magic characters for searching. However, you can use line feed in search patterns. For example, to find a line consisting of the three characters ‘‘foo’’ all by themselves on a line, using the mined bindings (see below), use the pattern: CTRL-\ CTRL-J f o o CTRL-\ CTRL-J. The CTRL- means to interpret the next character literally, in this case it is CTRL-J, which is line feed. You can also search for patterns involving multiple lines. For example, to find a line ending in an ‘‘x’’ followed by a line beginning with a ‘‘y’’, use as pattern: x CTRL- CTRL-J y.

Mined Key Bindings

These are the key bindings if the binary user profile, *.ellepro.b1*, is installed in $HOME. The ESCAPE key followed by a number followed by a command causes that command to be executed ‘‘number’’ times. This applies both to control characters and insertable characters. CTRL-X refers to a ‘‘control character.’’ ESC x refers to an escape character followed by x. In other words, ˆX is a synonym for

CTRL-X. ˆX Y refers to CTRL-X followed by y. To abort the current command and go back to the main loop of the editor, type CTRL-G, rather than CTRL-\.

Only a few commands are of the form CTRL-X Y. All of these are also bound to CTRL-X CTRL-Y, so you can hold down CTRL and then hit X Y, or release control after the X, as you prefer.

The key bindings that are not listed should not be used. Some of them actually do things. For example, the ANSI escape codes ESC [x are bound to ˆX Y for a variety of y.

Some commands work on regions. A region is defined as the text between the most recently set mark and the cursor.

Mined Commands

If the *mined* profile, *.ellepro.b1* is installed in your home directory, the following commands will work.

CURSOR MOTION

arrows	Move the cursor in the indicated direction
CTRL-A	Move cursor to start of current line
CTRL-Z	Move cursor to end of current line
CTRL-F	Move cursor forward word
CTRL-B	Move cursor backward to start of previous word

SCREEN MOTION

Home key	Move to first character of the file
End key	Move to last character of the file
PgUp key	Scroll window up 22 lines (closer to start of the file)
PgDn key	Scroll window down 22 lines (closer to end of the file)
CTRL-U	Scroll window up 1 line
CTRL-D	Scroll window down 1 line
ESC ,	Move to top of screen
CTRL-_	Move to bottom of screen

MODIFYING TEXT

DEL key	Delete the character under the cursor
Backsp	Delete the character to left of the cursor
CTRL-N	Delete the next word
CTRL-P	Delete the previous word
CTRL-T	Delete tail of line (all characters from cursor to end of line)

CTRL-O	Open up the line (insert line feed and back up)
ESC G	Get and insert a file at the cursor position (CTRL-G in mined)

REGIONS

CTRL-ˆ	Set mark at current position for use with CTRL-C and CTRL-K
CTRL-C	Copy the text between the mark and the cursor into the buffer
CTRL-K	Delete text between mark and cursor; also copy it to the buffer
CTRL-Y	Yank contents of the buffer out and insert it at the cursor

MISCELLANEOUS

numeric +	Search forward (prompts for expression)
numeric –	Search backward (prompts for expression)
CTRL-]	ESC n CTRL-[goes to line n (slightly different syntax than mined)
CTRL-R	Global replace pattern with string (from cursor to end)
CTRL-L	Replace pattern with string within the current line only
CTRL-W	Write the edited file back to the disk
CTRL-S	Fork off a shell (use CTRL-D to get back to the editor)
CTRL-G	Abort whatever the editor was doing and wait for command (CTRL-)
CTRL-E	Redraw screen with cursor line positioned in the middle
CTRL-V	Visit (edit) a new file
CTRL-Q	Write buffer to a file
ESC X	Exit the editor

Non-Mined Commands

CURSOR MOTION

ESC P	Forward paragraph (a paragraph is a line beginning with a dot)
ESC]	Backward paragraph
ESC .	Indent this line as much as the previous one

MODIFYING TEXT

CTRL-\	Insert the next character (used for inserting control characters)
ESC T	Transpose characters

ESC W	Transpose words
ESC =	Delete white space (horizontal space)
ESC |	Delete blank lines (vertical space)

REGIONS

ESC M	Mark current paragraph
ESC ˆ	Exchange cursor and mark
ESC Y	Yank back the next-to-the-last kill (CTRL-Y yanks the last one)
ESC A	Append next kill to kill buffer

KEYBOARD MACROS

ESC /	Start Keyboard Macro
ESC \	End Keyboard Macro
ESC *	View Keyboard Macro (the PrtSc key on the numeric pad is also a *)
ESC E	Execute Keyboard Macro

WINDOW MANAGEMENT

ˆX 1	Enter one window mode
ˆX 2	Enter two window mode
ˆX L	Make the current window larger
ˆX P	Make the window more petit/petite (Yes, Virginia, they are English)
ˆX N	Next window
ˆX W	New window

BUFFER MANAGEMENT

numeric 5	Display the list of current files and buffers
ESC B	Select a buffer
ESC S	Select an existing buffer
ESC N	Mark a buffer as NOT modified (even if it really is)

UPPER AND LOW CASE MANIPULATION

ESC I	Set first character of word to upper case
ESC C	Capitalize current word
ESC O	Make current word ordinary (i.e., lower case)
ESC U	Set entire region between mark and cursor to upper case
ESC L	Set entire region between mark and cursor to lower case

MISCELLANEOUS

ESC F	Find file and read it into its own buffer
ESC Z	Incremental search

ESC Q	Like CTRL-R, but queries at each occurrence (type ? for options)
ESC R	Reset the user profile from a file
ESC H	Help (ELLE prompts for the 1 or 2 character command to describe)
ESC ;	Insert a comment in a C program (generates /* */ for you)
ˆX X	Exit the editor (same as ESC X and CTRL-X CTRL-X)

The major differences between ELLE with the *mined* profile and *mined* itself are:

1. The definition of a ‘‘word’’ is different for forward and backward word
2. The mark is set with CTRL-ˆ instead of CTRL-@
3. Use CTRL-G to abort a command instead of CTRL-\
4. Use CTRL- to literally insert the next character, instead of ALT
5. CTRL-E adjusts the window to put the cursor in the middle of it
6. To get and insert a file, use ESC G instead of CTRL-G
7. To go to line n, type ESC n CTRL-[instead of CTRL-[n
8. You exit with CTRL-X CTRL-X and then answer the question with ‘‘y’’.
9. There are many new commands, windows, larger files, etc.

Emacs Key Bindings

If you do not have the *mined* profile installed, you get the standard Emacs key bindings. These are listed below. Commands not listed are not implemented.

CURSOR MOVEMENT

CTRL-F	Forward one character.
CTRL-B	Backward one character.
CTRL-H	Same as CTRL-B: move backward one character.
ESC F	Forward one word.
ESC B	Backward one word.
CTRL-A	Beginning of current line.
CTRL-E	End of current line.
CTRL-N	Next line (goes to the next line).
CTRL-P	Previous line (goes to the previous line).
CTRL-V	Beginning of next screenful.
ESC V	Beginning of previous screenful.
ESC]	Forward Paragraph.
ESC [	Backward Paragraph.
ESC <	Beginning of whole buffer.
ESC >	End of whole buffer.

DELETING

CTRL-D	Deletes forward one character (the one the cursor is under).
DELETE	Deletes backward one character (the one to left of cursor).
ESC D	Kills forward one word.
ESC DEL	Kills backward one word.
CTRL-K	Kills the rest of the line (to the right of the cursor).
ESC \	Deletes spaces around the cursor.
ˆX CTRL-O	Deletes blank lines around the cursor.

CASE CHANGE

ESC C	Capitalizes word : first letter becomes uppercase; rest lower
ESC L	Makes the whole next word lowercase.
ESC U	Makes the whole next word uppercase.
ˆX CTRL-L	Makes whole region lowercase.
ˆX CTRL-U	Makes whole region uppercase.

SEARCHING (If no string is given, previous string is used)

CTRL-S	Incremental Search forward; prompts ‘‘I-search:’’
CTRL-R	Reverse Incremental Search; prompts ‘‘R-search:’’

During an incremental search, the following characters have special effects:

‘‘normal’’	-	Begin searching immediately.
ˆG	-	Cancel I-search, return to start.
DEL	-	Erase last char, return to last match.
ˆS, ˆR	-	Repeat search (or change direction).
ESC or CR	-	Exit I-search at current point.

ESC %	Query Replace. Interactive replace. Type ‘‘?’’ to see options.
ˆX %	Replace String.Like Query Replace, but not interactive

MARKING AREAS

CTRL-ˆ	Set mark
ˆX CTRL-X	Exchange cursor and mark.
ESC H	Mark Paragraph. Sets mark and cursor to surround a para.
CTRL-W	Wipe-out -- kills a ‘‘region’’:
ESC W	Copy region. Like CTRL-W then CTRL-Y but modifies buffer
CTRL-Y	Yanks-back (un-kills) whatever you have most recently killed.

ESC Y	Yanks-back (un-kills) the next most recently killed text.
ESC CTRL-W	Append Next Kill. Accumulates stuff from several kills

FILLING TEXT

ESC Q	Fill the paragraph to the size of the Fill Column.
ESC G	Fill the region.
ˆX F	Set Fill Column. ESC Q will use this line size.
ˆX .	Set Fill Prefix. Asks for prefix string
ˆX T	Toggles Auto Fill Mode.

WINDOWS

ˆX 2	Make two windows (split screen).
ˆX 1	Make one window (delete window) (make one screen).
ˆX O	Go to Other window.
ˆX ˆ	Grow window: makes current window bigger.

BUFFERS

ˆX CTRL-F	Find a file and make a buffer for it.
ˆX B	Select Buffer: goes to specified buffer or makes new one
ˆX CTRL-B	Show the names of the buffers used in this editing session.
ˆX K	Kill Buffer.
ESC tilde	Say buffer is not modified.
ˆX CTRL-M	Toggle EOL mode (per-buffer flag).

KEYBOARD MACRO

ˆX (	Start collecting a keyboard macro.
ˆX)	Stop collecting.
ˆX E	Execute the collected macro.
ˆX *	Display the collected macro.

FILES

ˆX CTRL-I	Insert a file where cursor is.
ˆX CTRL-R	Read a new file into current buffer.
ˆX CTRL-V	Same as ˆX ˆR above (reads a file).
ˆX CTRL-W	Write buffer out to new file name.
ˆX CTRL-S	Save file: write out buffer to its file name.
ˆX CTRL-E	Write region out to new file name.

MISCELLANEOUS

ˆX CTRL-Z	Exit from ELLE.
ˆX !	Escape to shell (CTRL-D to return)
CTRL-O	Open up line

LINEFEED	Same as typing RETURN and TAB.
CTRL-T	Transposes characters.
ESC T	Transposes words.
CTRL-U	Makes the next command happen four times.
CTRL-U number	Makes the next command happen ‘‘number’’ times.
ESC number	Same as CTRL-U number.
CTRL-L	Refreshes screen.
CTRL-U CTRL-L	Refresh only the line cursor is on.
CTRL-U n CTRL-L	Change window so the cursor is on line n
CTRL-Q	Quote: insert the next character no matter what it is.
CTRL-G	Quit: use to avoid answering a question.
ESC ;	Inserts comment (for writing C programs).
ESC I	Inserts indentation equal to previous line.
ESC M	Move to end of this line’s indentation.
CTRL-_	Describe a command (if the command database is online)

UNUSED CONTROLS

CTRL-C	Not used.
CTRL-Z	Not used.
CTRL-]	Not used.

ELLE profile

It is possible to create your own user profile. The mechanism is different from Emacs, since ELLE does not have Mock Lisp. Proceed as follows.

1. Modify *.ellepro.e* to suit your taste.
2. Install *.ellepro.e* in your home directory.
3. Type:

   ```
   ellec –Profile
   ```

4. Check to see if *.ellepro.b1* has been created. If it has, you are ready to go.

Author

ELLE was written by Ken Harrenstien of SRI (klh@sri.com).